Chapter 10. Cluster Analysis: Basic Concepts and Methods

What is Cluster Analysis?

- Cluster: A collection of data objects
  - similar (or related) to one another within the same group
  - dissimilar (or unrelated) to the objects in other groups
- Cluster analysis (or clustering, data segmentation, ...)
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters
- Unsupervised learning: no predefined classes (i.e., learning by observations vs. learning by examples: supervised)
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms
Applications of Cluster Analysis

- Data reduction
  - Summarization: Preprocessing for regression, PCA, classification, and association analysis
- Prediction based on groups
  - Cluster & find characteristics/patterns for each group
- Finding K-nearest Neighbors
  - Localizing search to one or a small number of clusters
- Outlier detection: Outliers are often viewed as those “far away” from any cluster

Clustering: Application Examples

- Biology: taxonomy of living things: kingdom, phylum, class, order, family, genus and species
- Information retrieval: document clustering
- Land use: Identification of areas of similar land use in an earth observation database
- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Climate: understanding earth climate, find patterns of atmospheric and ocean

Clustering: Application Examples

- Biology: Antigenic Cartography

Clustering: Application Examples

- Document Clustering
Land use: Identification of areas of similar land use in an earth observation database

Basic Steps to Develop a Clustering Task

- Feature selection
  - Select info concerning the task of interest
  - Minimal information redundancy
- Proximity measure
  - Similarity of two feature vectors
- Clustering criterion
  - Expressed via a cost function or some rules
- Clustering algorithms
  - Choice of algorithms
- Validation of the results
  - Validation test (also, clustering tendency test)
- Interpretation of the results
  - Integration with applications

Clustering: Application Examples

Marketing

Quality: What Is Good Clustering?

- A good clustering method will produce high quality clusters
  - high intra-class similarity: cohesive within clusters
  - low inter-class similarity: distinctive between clusters
- The quality of a clustering method depends on
  - the similarity measure used by the method
  - its implementation, and
  - Its ability to discover some or all of the hidden patterns
Measure the Quality of Clustering

- Dissimilarity/Similarity metric
  - Similarity is expressed in terms of a distance function, typically metric: $d(i, j)$
  - The definitions of distance functions are usually rather different for interval-scaled, boolean, categorical, ordinal ratio, and vector variables
  - Weights should be associated with different variables based on applications and data semantics
  - Quality of clustering:
    - There is usually a separate "quality" function that measures the "goodness" of a cluster.
    - It is hard to define "similar enough" or "good enough"
      - The answer is typically highly subjective

Considerations for Cluster Analysis

- Partitioning criteria
  - Single level vs. hierarchical partitioning (often, multi-level hierarchical partitioning is desirable)
- Separation of clusters
  - Exclusive (e.g., one customer belongs to only one region) vs. non-exclusive (e.g., one document may belong to more than one class)
- Similarity measure
  - Distance-based (e.g., Euclidian, road network, vector) vs. connectivity-based (e.g., density or contiguity)
- Clustering space
  - Full space (often when low dimensional) vs. subspaces (often in high-dimensional clustering)

Requirements and Challenges

- Scalability
  - Clustering all the data instead of only on samples
- Ability to deal with different types of attributes
  - Numerical, binary, categorical, ordinal, linked, and mixture of these
- Constraint-based clustering
  - User may give inputs on constraints
  - Use domain knowledge to determine input parameters
- Interpretability and usability
- Others
  - Discovery of clusters with arbitrary shape
  - Ability to deal with noisy data
  - Incremental clustering and insensitivity to input order
  - High dimensionality

Major Clustering Approaches (I)

- Partitioning approach:
  - Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors
  - Typical methods: k-means, k-medoids, CLARANS
- Hierarchical approach:
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Typical methods: UPGMA
- Density-based approach:
  - Based on connectivity and density functions
  - Typical methods: DBSCAN, OPTICS, DenClue
- Grid-based approach:
  - based on a multiple-level granularity structure
  - Typical methods: STING, WaveCluster, CLIQUE
Chapter 10. Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering
- Summary

Partitioning Algorithms: Basic Concept

- **Partitioning method**: Partitioning a database \( D \) of \( n \) objects into a set of \( k \) clusters, such that the sum of squared distances is minimized (where \( c_i \) is the centroid of cluster \( C_i \))

\[
E = \sum_{i=1}^{k} \sum_{p \in C_i} (d(p, c_i))^2
\]

- Given \( k \), find a partition of \( k \) clusters that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: \( k\)-means and \( k\)-medoids algorithms
  - \( k\)-means (MacQueen’67, Lloyd’57/’82): Each cluster is represented by the center of the cluster
  - \( k\)-medoids or PAM (Partition around medoids) (Kaufman & Rousseeuw’87): Each cluster is represented by one of the objects in the cluster

The \( K\)-Means Clustering Method

- Given \( k \), the \( k\)-means algorithm is implemented in four steps:
  - Partition objects into \( k \) nonempty subsets
  - Compute seed points as the centroids of the clusters of the current partitioning (the centroid is the center, i.e., mean point, of the cluster)
  - Assign each object to the cluster with the nearest seed point
  - Go back to Step 2, stop when the assignment does not change

An Example of \( K\)-Means Clustering
Within and Between Cluster Criteria

Let's consider total point scatter for a set of \( N \) data points:

\[
T = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} d(x_i, x_j)
\]

Distance between two points

\( T \) can be re-written as:

\[
T = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} |d(x_i, x_j)|
\]

Within cluster scatter

Between cluster scatter

Minimizing \( W(C) \) is equivalent to maximizing \( B(C) \)

Example

- **Problem**
  Suppose we have 4 types of medicines and each has two attributes (pH and weight index). Our goal is to group these objects into \( K=2 \) group of medicine.

<table>
<thead>
<tr>
<th>Medicine</th>
<th>Weight</th>
<th>pH-Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>

Example

- **Step 1:** Use initial seed points for partitioning

\[ c_1 = A, c_2 = B \]

\[ d(D, c_1) = \sqrt{(3-1)^2 + (4-1)^2} = 5 \]

\[ d(D, c_2) = \sqrt{(5-2)^2 + (4-1)^2} = 4.24 \]

Assign each object to the cluster with the nearest seed point

Example

- **Step 2:** Compute new centroids of the current partition

\[ c_2 = \left( \frac{2 + 4 + 5}{3}, \frac{1 + 3 + 4}{3} \right) \]

\[ = (\frac{11}{3}, \frac{8}{3}) \]

\[ = (3.67, 2.67) \]
Example

- Step 3: Repeat the first two steps until its convergence.

Knowing the members of each cluster, now we compute the new centroid of each group based on these new memberships.

\[ c_1 = \left( \frac{1 + 2}{2}, \frac{1 + 1}{2} \right) = (1, 1) \]

\[ c_2 = \left( \frac{4 + 5}{2}, \frac{3 + 4}{2} \right) = (4.5, 3.5) \]

Exercise

For the medicine data set, use K-means with the distance metric for clustering analysis by setting K=2 and initializing seeds as \( C_1 = A \) and \( C_2 = C \). Answer three questions as follows:
1. How many steps are needed for convergence?
2. What are memberships of two clusters?
3. What are centroids of two clusters after convergence?

<table>
<thead>
<tr>
<th>Medicine</th>
<th>Weight</th>
<th>pH-Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>

How K-mean partitions?

When \( K \) centroids are set/fixed, they partition the whole space into \( K \) subspaces constituting a partitioning.

Changing positions of centroids leads to a new partitioning.

A partitioning amounts to a Voronoi Diagram.
1. User set up the number of clusters they'd like. (e.g. \( k=5 \))

2. Randomly guess \( K \) cluster center locations

3. Each data point finds out which center it's closest to. (Thus each center "owns" a set of data points)

4. Each center finds the centroid of the points it owns
**K-means Demo**

1. User set up the number of clusters they'd like. (e.g. \(K=5\))
2. Randomly guess \(K\) cluster centres locations
3. Each data point finds out which centre it's closest to. (Thus each centre "owns" a set of data points)
4. Each centre finds the centroid of the points it owns
5. ...and jumps there

**Comments on the K-Means Method**

- **Strength:** Efficient: \(O(tk(n))\), where \(n\) is # objects, \(k\) is # clusters, and \(t\) is # iterations. Normally, \(k, t \ll n\).
  - Comparing: PAM: \(O(k(n-k)^2)\), CLARA: \(O(ks^2 + k(n-k))\)
- **Comment:** Often terminates at a local optimal
- **Weakness**
  - Applicable only to objects in a continuous \(n\)-dimensional space
  - In comparison, \(k\)-medoids can be applied to a wide range of data
  - Need to specify \(k\), the number of clusters, in advance (there are ways to automatically determine the best \(k\) (see Hastie et al., 2009)
  - Sensitive to noisy data and outliers
  - Not suitable to discover clusters with non-convex shapes

**Variations of the K-Means Method**

- Most of the variants of the \(k\)-means which differ in
  - Selection of the initial \(k\) means
  - Dissimilarity calculations
  - Strategies to calculate cluster means
What Is the Problem of the K-Means Method?

- The k-means algorithm is sensitive to outliers!
  - Since an object with an extremely large value may substantially distort the distribution of the data

- K-Medoids: Instead of taking the mean value of the object in a cluster as a reference point, medoids can be used, which is the most centrally located object in a cluster.

The K-Medoid Clustering Method

- K-Medoids Clustering: Find representative objects (medoids) in clusters
  - PAM (Partitioning Around Medoids, Kaufmann & Rousseeuw 1987)
    - Starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
    - PAM works effectively for small data sets, but does not scale well for large data sets (due to the computational complexity)
  - Efficiency improvement on PAM
    - CLARA (Kaufmann & Rousseeuw, 1990): PAM on samples
    - CLARANS (Ng & Han, 1994): Randomized re-sampling

PAM: A Typical K-Medoids Algorithm

Chapter 10. Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering
- Summary
Hierarchical Clustering

- Use distance matrix as clustering criteria. This method does not require the number of clusters $k$ as an input, but needs a termination condition.

AGNES (Agglomerative Nesting)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical packages, e.g., Splus
- Use the single-link method and the dissimilarity matrix
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster

Cluster Distance Measures

- Single link: smallest distance between an element in one cluster and an element in the other, i.e., $d(C_i, C_j) = \min(d(x_i, x_j))$
- Complete link: largest distance between an element in one cluster and an element in the other, i.e., $d(C_i, C_j) = \max(d(x_i, x_j))$
- Average (UPGMA: Unweighted Pair Group Method with Arithmetic Mean): Avg distance between elements in one cluster and elements in the other, i.e., $d(C_i, C_j) = \text{avg}(d(x_i, x_j))$
## Cluster Distance Measures

**Example:** Given a data set of five objects characterized by a single feature, assume that there are two clusters: \( C_1: \{a, b\} \) and \( C_2: \{c, d, e\} \).

<table>
<thead>
<tr>
<th>Feature</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

1. Calculate the distance matrix.

2. Calculate three cluster distances between \( C_1 \) and \( C_2 \).

- **Single link**
  \[
  \text{dist}(C_1, C_2) = \min\{d(a,c), d(a,d), d(a,e), d(b,c), d(b,d), d(b,e)\} = \min\{3, 4, 5\} = 3
  \]

- **Complete link**
  \[
  \text{dist}(C_1, C_2) = \max\{d(a,c), d(a,d), d(a,e), d(b,c), d(b,d), d(b,e)\} = \max\{3, 4, 5\} = 5
  \]

- **Average**
  \[
  \text{dist}(C_1, C_2) = \frac{d(a,c) + d(a,d) + d(a,e) + d(b,c) + d(b,d) + d(b,e)}{6} = \frac{3 + 4 + 5 + 4 + 4}{6} = \frac{21}{6} = 3.5
  \]

### Problem: clustering analysis with agglomerative algorithm

#### Example data matrix

#### Distance matrix

#### Example

- **Merge two closest clusters (iteration 1)**
Example

- **Update distance matrix (iteration 1)**
  - \[ d_{\text{A,B}} = \min(d_{\text{A,A}}, d_{\text{A,B}}) = \min(5.66, 4.95) = 4.95 \]
  - \[ d_{\text{B,C}} = \min(d_{\text{B,B}}, d_{\text{B,C}}) = \min(2.24, 0.71) = 0.71 \]
  - \[ d_{\text{C,D}} = \min(d_{\text{C,C}}, d_{\text{C,D}}) = \min(4.95, 2.24) = 2.24 \]
  - \[ d_{\text{D,E}} = \min(d_{\text{D,D}}, d_{\text{D,E}}) = \min(3.20, 4.24) = 3.20 \]

Example

- **Merge two closest clusters (iteration 2)**

Example

- **Update distance matrix (iteration 2)**

Example

- **Merge two closest clusters/update distance matrix (iteration 3)**
Example

- Merge two closest clusters/update distance matrix (iteration 4)

<table>
<thead>
<tr>
<th>Min Distance (Single Linkage)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A, B)</td>
</tr>
<tr>
<td>0.00</td>
</tr>
<tr>
<td>4.95</td>
</tr>
<tr>
<td>2.50</td>
</tr>
</tbody>
</table>

Example

- Final result (meeting termination condition)

<table>
<thead>
<tr>
<th>X1</th>
<th>X2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>1.5</td>
</tr>
<tr>
<td>C</td>
<td>5</td>
</tr>
<tr>
<td>D</td>
<td>3</td>
</tr>
<tr>
<td>E</td>
<td>4</td>
</tr>
<tr>
<td>F</td>
<td>3</td>
</tr>
</tbody>
</table>

Chapter 10. Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering
- Summary

Limitations of K-means: Non-globular Shapes

Original Points

K-means (2 Clusters)
Density-Based Clustering

- Clustering based on density (local cluster criterion), such as density-connected points
- Each cluster has a considerably higher density of points than outside of the cluster

Density-Based Clustering Methods

- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters as termination condition
- Several interesting studies:
  - **DBSCAN**: Ester, et al. (KDD’96)
  - **DENCLUE**: Hinneburg & D. Keim (KDD’98)
  - **CLIQUE**: Agrawal, et al. (SIGMOD’98) (more grid-based)

Density-Based Clustering: Basic Concepts

- Two parameters:
  - \( \epsilon \): Maximum radius of the neighborhood
  - MinPts: Minimum number of points in an \( \epsilon \)-neighborhood of that point
- \( N(p) = \{ q \in D \mid \text{dist}(p,q) \leq \epsilon \} \)
- **Density**: \# of objects in the neighborhood
  \[
  \text{density}(p) = |N(p)|
  \]
  \( \text{density}(p) = 5 \)
  \( \text{density}(q) = 9 \)
Density-Based Clustering: Basic Concepts

- Two parameters:
  - \( \varepsilon \): Maximum radius of the neighborhood
  - MinPts: Minimum number of points in an \( \varepsilon \)-neighborhood of that point

- Directly density-reachable: A point \( p \) is directly density-reachable from a point \( q \) w.r.t. \( \varepsilon \), MinPts if
  - \( p \) belongs to \( N_\varepsilon(q) \)
  - Core point condition:

Density-Reachable and Density-Connected

- Density-reachable:
  - A point \( p \) is density-reachable from a point \( q \) w.r.t. \( \varepsilon \), MinPts if there is a chain of points \( p_1, \ldots, p_n \) such that \( p_1 = q \), \( p_n = p \) is directly density-reachable from \( p_i \)

- Density-connected

Density-Connectivity

- Density-connected
  - A point \( p \) is density-connected to a point \( q \) w.r.t. \( \varepsilon \), MinPts if there is a point \( o \) such that both \( p \) and \( q \) are density-reachable from \( o \) w.r.t. \( \varepsilon \) and MinPts
Density-Based Clustering: Basic Concepts

- **Core point:**
  \[ \text{density}(p) = |N_{\varepsilon}(p)| \geq \text{MinPts} \]
  \[ N_{\varepsilon}(p) = \{ q \in D | \text{dist}(p, q) \leq \varepsilon \} \]

- DBSCAN: Density-Based Spatial Clustering of Applications with Noise
  - Relies on a density-based notion of cluster:
    - A cluster is defined as a maximal set of density-connected points.
  - Discovers clusters of arbitrary shape in spatial databases with noise.

Density-Based Cluster

- A subset \( C \subset D \) is a cluster if:
  1. Any two objects \( p \in C \) and \( q \in C \) are density-connected.
  2. Any two objects \( p \in C \) and \( q \in D \setminus C \) are NOT density-connected.

DBSCAN: The Algorithm

1. Arbitrarily select a point \( p \).
2. Retrieve all points density-reachable from \( p \) w.r.t. \( \varepsilon \) and \( \text{MinPts} \).
3. If \( p \) is a core point, a cluster is formed; Add \( N_{\varepsilon}(p) \) to the set \( N \).
4. If \( p \) is a border point, no points are density-reachable from \( p \) and DBSCAN visits the next point of the database.
5. Continue the process until all of the points have been processed.
**DBSCAN**: The Algorithm

Algorithm: **DBSCAN**: a density-based clustering algorithm.

**Input:**
- \( D \): a data set containing \( n \) objects,
- \( \varepsilon \): the radius parameter, and
- \( MinPts \): the neighborhood density threshold.

**Output:** A set of density-based clusters.

---

**DBSCAN**: Experimental Results

*DBSCAN* can find non-linearly separable clusters

*K-means* does not work

---

**DBSCAN**: The Algorithm

1. mark all objects as unvisited.
2. do
3. randomly select an unvisited object \( p \).
4. mark \( p \) as visited.
5. if the \( \varepsilon \)-neighborhood of \( p \) has at least \( MinPts \) objects
6. create a new cluster \( C \), and add \( p \) to \( C \).
7. let \( N \) be the set of objects in the \( \varepsilon \)-neighborhood of \( p \).
8. for each point \( p' \) in \( N \)
9. if \( p' \) is unvisited
10. mark \( p' \) as visited.
11. if the \( \varepsilon \)-neighborhood of \( p' \) has at least \( MinPts \) points, add those points to \( N \).
12. if \( p' \) is not yet a member of any cluster, add \( p' \) to \( C \).
13. end for
14. output \( C \).
15. else mark \( p \) as noise.
16. until no object is unvisited.
**DBSCAN : Sensitive to Parameters**

Input Dataset

Clustering Results

MinPts = 4
\[ \epsilon = 3.5 \]

**Advantages**

- DBSCAN does not require one to specify the number of clusters.
- DBSCAN can find arbitrarily shaped clusters. It can even find a cluster completely surrounded by (but not connected to) a different cluster.
- DBSCAN requires just two parameters and is mostly insensitive to the ordering of the points in the database.

**Disadvantages**

- Sensitive to Parameters
- DBSCAN cannot cluster data sets well with large differences in densities

**OPTICS: A Cluster-Ordering Method**

OPTICS: Ordering Points To Identify the Clustering Structure

- Ankerst, Breunig, Kriegel, and Sander (SIGMOD’99)
- Produces a special order of the database wrt its density-based clustering structure
- This cluster-ordering contains info equiv to the density-based clusterings corresponding to a broad range of parameter settings
- Good for both automatic and interactive cluster analysis, including finding intrinsic clustering structure
- Can be represented graphically or using visualization techniques
**OPTICS: Some Extension from DBSCAN**

Core-Distance$_{	ext{MinPts}}(p) = \begin{cases} \text{Undefined} & \text{if } |X_c(p)| < \text{MinPts} \\ \epsilon' & \text{if } |X_c(p)| \geq \text{MinPts} \end{cases}

$\epsilon'$: minimum distance threshold that makes $p$ a core object

- $\epsilon = 6$ mm
- MinPts = 5
- $\epsilon' = 3$ mm

Reachability-Distance$_{	ext{MinPts},q}(p) = \begin{cases} \text{Undefined} & \text{if } |X_c(q)| < \text{MinPts} \\ \max(\epsilon', \text{dist}(p, q)) & \text{if } |X_c(q)| \geq \text{MinPts} \end{cases}

$\epsilon'$: minimum distance threshold that makes $q$ a core object

- $\epsilon = 6$ mm
- MinPts = 5
- $\epsilon' = 3$ mm
- $p': \text{dist}(p', q)$

**Cluster Order**

OPTICS does not directly return a (hierarchical) clustering, but orders the objects according to a "cluster order" w.r.t. $\epsilon$ and MinPts.

Cluster order w.r.t. $\epsilon$ and MinPts:
- start with an arbitrary object
- visit the object that has the minimum reachability distance from the set of already visited objects

**OPTICS- Example**

- Example Database (2-dimensional, 16 points)
- $\epsilon = 44$, MinPts = 3

seedlist: (B, 40) (I, 40)
OPTICS Example

- Example Database (2-dimensional, 16 points)
  - $\varepsilon = 44$, $MinPts = 3$

seedlist: (I, 40) (C, 40)

OPTICS Example

- Example Database (2-dimensional, 16 points)
  - $\varepsilon = 44$, $MinPts = 3$

seedlist: (J, 20) (K, 20) (L, 31) (C, 40) (M, 40) (R, 43)

OPTICS Example

- Example Database (2-dimensional, 16 points)
  - $\varepsilon = 44$, $MinPts = 3$

seedlist: (L, 19) (K, 20) (R, 21) (M, 30) (P, 31) (C, 40)

OPTICS Example

- Example Database (2-dimensional, 16 points)
  - $\varepsilon = 44$, $MinPts = 3$

seedlist: (M, 18) (K, 18) (R, 20) (P, 21) (N, 35) (C, 40)
OPTICS- Example

- Example Database (2-dimensional, 16 points)
  - $\varepsilon = 44$, $MinPts = 3$

Reachability Diagram

- Depicts the reachability distances (w.r.t. $\varepsilon$ and $MinPts$) of all objects in a bar diagram
- With the objects ordered according to the cluster order

Sensitivity of Parameters

- Cluster order is robust against changes of the parameters good results as long as parameters “large enough”
Manual Cluster Analysis

Based on Reachability Diagram
- Are there clusters?
- How many clusters?
- How large are the clusters?
- Are the clusters hierarchically nested?

Reachability Diagram

Chapter 10. Cluster Analysis: Basic Concepts and Methods
- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering
- Summary

Automatic Cluster Analysis

ε-Cluster
- Subsequence of the cluster order
- Starts in an area of ε-steep decreasing reachability distances
- Ends in an area of ε-steep increasing reachability distances at approximately the same absolute value
- Contains at least MinPts objects

Algorithm
- Determines all ε-clusters
- Marks the ε-clusters in the reachability diagram
- Runtime complexity $O(n)$

Grid-Based Clustering Method
- Using multi-resolution grid data structure
- Several interesting methods
  - CLIQUE: Agrawal, et al. (SIGMOD'98)
    - Both grid-based and subspace clustering
  - STING (a STatistical INformation Grid approach) by Wang, Yang and Muntz (1997)
  - WaveCluster by Sheikholeslami, Chatterjee, and Zhang (VLDB'98)
    - A multi-resolution clustering approach using wavelet method
Graph-based models

- Clique i.e., a subset of nodes in a graph such that every two nodes in the subset are connected by an edge can be considered as a prototypical form of cluster.

Corrupted Cliques Problem

**Input:** A graph $G$

**Output:** The smallest number of additions and removals of edges that will transform $G$ into a clique graph

Distance Graphs

- Turn the distance matrix into a distance graph
  - Genes are represented as vertices in the graph
  - Choose a distance threshold $\theta$
  - If the distance between two vertices is below $\theta$, draw an edge between them
  - The resulting graph may contain cliques
  - These cliques represent clusters of closely located data points!

Community Detection
Community Detection in Graphs

What is a "Community"?
- Densely connected inside
- Sparsely connected outside

<table>
<thead>
<tr>
<th>Inside</th>
<th>Outside</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dense</td>
<td>Sparse</td>
</tr>
<tr>
<td>Cohesive</td>
<td>Loosely connected</td>
</tr>
</tbody>
</table>

Twitter Follower Network

Co-Author Network

Basketball
Football
Finding Community Structures

- Divide the network into non-empty groups (communities) in such a way that every vertex belongs to one of the communities.
- Many possible divisions could be done.
- We need a good division.
- Measurement of good division.

Finding Community Structure in very large networks

- Consider edges that fall within a community or between a community and the rest of the network.
- Define modularity:
  \[
  Q = \frac{1}{2m} \sum_{vw} \left( A_{vw} - \frac{k_v k_w}{2m} \right) \delta(c_v, c_w)
  \]
  where:
  - \( A_{vw} \) is the adjacency matrix
  - \( k_v \) and \( k_w \) are the degrees of vertices \( v \) and \( w \)
  - \( \delta(c_v, c_w) \) is 1 if vertices are in the same community, 0 otherwise
  - The probability of an edge between two vertices is proportional to their degrees. The expected number of edges between two nodes is proportional to their degrees.
  - For a random network, \( Q = 0 \)
  - The number of edges within a community is no different from what you would expect
  - \( 0.3 < Q < 0.7 \) is significant community structure. Greedy approach to maximize \( Q \).

Algorithm
- Start with all vertices as isolates
- Follow a greedy strategy:
  - SUCCESSIVELY join clusters with the greatest increase \( \Delta Q \) in modularity
  - Stop when the maximum possible \( \Delta Q \leq 0 \) from joining any two
- Successfully used to find community structure in a graph with > 400,000 nodes with > 2 million edges
- Amazon’s people who bought this also bought that...

Subspace Models

- Biclustering (also known as Co-clustering or two-mode-clustering), clusters are modeled with both cluster members and relevant attributes.
Biclustering

- Microarray data can be viewed as an N×M matrix:
  - Each of the N rows represents a gene (or a clone, ORF, etc.).
  - Each of the M columns represents a condition (a sample, a time point, etc.).
  - Each entry represents the expression level of a gene under a condition. It can either be an absolute value (e.g. Affymetrix GeneChip) or a relative expression ratio (e.g. cDNA microarrays).
  - A row/column is sometimes referred to as the "expression profile" of the gene/condition.

Biclustering

- As a result, each cluster may involve only a subset of genes and a subset of conditions, which form a "checkerboard" structure:

In reality, each gene/condition may participate in multiple clusters.

Biclustering

- Different biclustering algorithms have different definitions of bicluster

<table>
<thead>
<tr>
<th>a) Bicluster with constant values</th>
<th>b) Bicluster with constant values on rows</th>
<th>c) Bicluster with constant values on columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0 2.0 2.0 2.0 2.0</td>
<td>1.0 1.0 1.0 1.0 1.0</td>
<td>1.0 2.0 3.0 4.0 5.0</td>
</tr>
<tr>
<td>2.0 2.0 2.0 2.0 2.0</td>
<td>2.0 2.0 2.0 2.0 2.0</td>
<td>1.0 2.0 3.0 4.0 5.0</td>
</tr>
<tr>
<td>2.0 2.0 2.0 2.0 2.0</td>
<td>3.0 3.0 3.0 3.0 3.0</td>
<td>1.0 2.0 3.0 4.0 5.0</td>
</tr>
<tr>
<td>2.0 2.0 2.0 2.0 2.0</td>
<td>4.0 4.0 4.0 4.0 4.0</td>
<td>1.0 2.0 3.0 4.0 5.0</td>
</tr>
<tr>
<td>2.0 2.0 2.0 2.0 2.0</td>
<td>4.0 4.0 4.0 4.0 4.0</td>
<td>1.0 2.0 3.0 4.0 5.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>d) Bicluster with coherent values (additive)</th>
<th>e) Bicluster with coherent values (multiplicative)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0 4.0 5.0 0.0 1.5</td>
<td>1.0 0.6 2.0 0.2 0.8</td>
</tr>
<tr>
<td>4.0 7.0 8.0 3.0 4.5</td>
<td>2.0 1.0 4.0 0.4 1.6</td>
</tr>
<tr>
<td>3.0 6.0 7.0 2.0 3.6</td>
<td>3.0 1.5 6.0 0.6 2.4</td>
</tr>
<tr>
<td>5.0 6.0 9.0 4.0 5.5</td>
<td>4.0 2.0 8.0 0.8 3.2</td>
</tr>
<tr>
<td>2.0 6.0 6.0 1.0 2.6</td>
<td>6.0 2.6 10.0 1.0 4.0</td>
</tr>
</tbody>
</table>
**Chapter 10. Cluster Analysis: Basic Concepts and Methods**

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering
- Summary

---

**Determine the Number of Clusters**

- **Empirical method**
  - # of clusters: \( k = \sqrt{n/2} \) for a dataset of \( n \) points, e.g., \( n = 200, k = 10 \)
- **Cross validation method**
  - Divide a given data set into \( m \) parts
  - Use \( m - 1 \) parts to obtain a clustering model
  - Use the remaining part to test the quality of the clustering
  - E.g., For each point in the test set, find the closest centroid, and use the sum of squared distance between all points in the test set and the closest centroids to measure how well the model fits the test set
  - For any \( k > 0 \), repeat it \( m \) times, compare the overall quality measure w.r.t. different \( k \)'s, and find # of clusters that fits the data the best

---

**Measuring Clustering Quality**

- 3 kinds of measures: External, internal and relative
- External: supervised, employ criteria not inherent to the dataset
  - Compare a clustering against prior or expert-specified knowledge (i.e., the ground truth) using certain clustering quality measure
- Internal: unsupervised, criteria derived from data itself
  - Evaluate the goodness of a clustering by considering how well the clusters are separated, and how compact the clusters are.
  - Relative: directly compare different clusterings, usually those obtained via different parameter settings for the same algorithm
Measuring Clustering Quality: External Methods

- Clustering quality measure: $Q(C, C_g)$, for a clustering $C$ given the ground truth $C_g$.
- $Q$ is good if it satisfies the following 4 essential criteria:
  - Cluster homogeneity: the purer, the better
  - Cluster completeness: should assign objects belong to the same category in the ground truth to the same cluster
  - Rag bag: putting a heterogeneous object into a pure cluster should be penalized more than putting it into a rag bag.
  - Small cluster preservation: splitting a small category into pieces is more harmful than splitting a large category into pieces
Small cluster preservation

Internal: Silhouette coefficient

The silhouette coefficient is such a measure. For a data set, \( D \), of \( n \) objects, suppose \( D \) is partitioned into \( k \) clusters, \( C_1, \ldots, C_k \). For each object \( o \in D \), we calculate \( a(o) \) as the average distance between \( o \) and all other objects in the cluster to which \( o \) belongs. Similarly, \( b(o) \) is the minimum average distance from \( o \) to all clusters to which \( o \) does not belong. Formally, suppose \( o \in C_i \) (\( 1 \leq i \leq k \)), then

\[
a(o) = \frac{\sum_{o' \in C_i, o' \neq o} \text{dist}(o, o')}{|C_i| - 1}
\]

and

\[
b(o) = \min_{C_j \neq C_i} \left\{ \frac{\sum_{o' \in C_j} \text{dist}(o, o')}{|C_j|} \right\}.
\]

The silhouette coefficient of \( o \) is then defined as

\[
s(o) = \frac{b(o) - a(o)}{\max(a(o), b(o))}.
\]

Chapter 10. Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering
- Summary

Summary

- Cluster analysis groups objects based on their similarity and has wide applications
- Measure of similarity can be computed for various types of data
- Clustering algorithms can be categorized into partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods
- K-means and K-medoids algorithms are popular partitioning-based clustering algorithms
- Birch and Chameleon are interesting hierarchical clustering algorithms, and there are also probabilistic hierarchical clustering algorithms
- DBSCAN, OPTICS, and DEENCU are interesting density-based algorithms
- STING and CLIQUE are grid-based methods, where CLIQUE is also a subspace clustering algorithm
- Quality of clustering results can be evaluated in various ways